

REMARKS

I. Introduction

In response to the Office Action dated February 25, 2004, claim 3 has been amended. Claims 1-27 remain in the application. Re-examination and re-consideration of the application, as amended, is requested.

II. Prior Art Rejections

In paragraph (4) of the Office Action, claims 1-3, 5-11, 13-19, and 21-27 were rejected under 35 U.S.C. §103(a) as being unpatentable over Ko et al., U.S. Patent No. 6,292,185 B1 (Ko) in view of Kasso et al., U.S. Patent No. 5,832,505 (Kasso). In paragraph (5) of the Office Action, claims 4, 12, and 20 were rejected under 35 U.S.C. §103(a) as being unpatentable over Ko in view of Kasso as applied to claim 1, and further in view of Estrada et al., U.S. Patent No. 6,594,664 (Estrada).

Applicants respectfully traverse these rejections.

Specifically, claims 1, 9, 17, and 25-27 were rejected as follows:

Regarding claims 1, 9, 17, & 25-27 Ko discloses a computer-implemented method (Col. 9: 24-37), a system (FIG. 1) and an article of manufacture (Col. 10:1-48) for electronically obtaining a Web page in a Web browser comprising:
requesting a Web page (FIG.2, 201); a script block comprising a method that accesses the component module control object (Col.4:46-50, see Visual Basic Scripting); obtaining the component module control object, wherein the component module control object comprises one or more resources (Col.4:30-33, see request to download object); installing the component module control object (FIG.3,304); and using the method of the script block to extract a Web page based on the resources of the component module control object (Col.4:57-63, see if not native, and directive in CODEBASE, to download contained in the default.htm). Ko doesn't explicitly disclose obtaining a bootstrap file, wherein the bootstrap file comprises a declaration of a component module control object. However, Kasso does disclose this feature in analogous prior art (Col.5:20-35, section BOOT SEQUENCE). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Ko with Kasso to implement the instant claimed invention because, obtaining a bootstrap file would enable an operating system to be executed or loaded from a remote location (Col.4:54-60).

Applicants traverse the above rejections for one or more of the following reasons:

- (1) Neither Ko, Kasso, or Estrada teach, disclose or suggest a bootstrap file comprising a declaration of a component module control object and a script block;
- (2) Neither Ko, Kasso, or Estrada teach, disclose or suggest a component module control object that comprises resources; and

(3) Neither Ko, Kasso, or Estrada teach, disclose or suggest using a method of a script block to extract a Web page based on the resources of the component module control object.

Independent claims 1, 9, 17, and 25-27 are generally directed to creating web pages. Specifically, a Web browser requests a Web page from a server. In response, a bootstrap file is received by the Web browser. The claims specifically provide that the bootstrap file comprises a declaration of a component module control object and a script block that has a method that accesses the object. The Web browser then obtains the control object. It should be noted that the claimed control object consists of one or more resources. Once obtained, the control object is installed and a method of the script block is used to extract a Web page based on the resources in the control object.

The cited references do not teach nor suggest these various elements of Applicants' independent claims.

Ko merely describes a method and apparatus for tailoring the appearance of a graphical user interface, specifically an internet web browser. A host server receives a request over the internet or an intranet for a web page. The host server provides the necessary data and executable files for the user's personal computer. A computer program executed on a personal computer alters the appearance of a user's graphical web browser by accessing data files. The appearance of a web browser can be tailored using an editor program to provide data files used to tailor the appearance of a web browser.

The Office Action admits that Ko fails to teach a bootstrap file. To teach this element of the claim, the Office Action relies on Kasso, col. 5, lines 20-35. Firstly, Applicants again note that the claimed bootstrap file has a declaration of a component module control object AND a script block that has a method that access the control object. In this regard, to teach the bootstrap file, Kasso must teach both the declaration of the component module control object and a script block that has a method that accesses the control object.

Instead of teaching these elements, Kasso merely describes a sequence of steps performed by a computer when the computer is powered up (see col. 5, lines 20-22). In this regard, a boot procedure searches for a JavaOS boot server, establishes a network connection with the server, and downloads an operating system from the server into the RAM of the host. The host also downloads an application program that is embedded in a boot image. The boot image also includes parameters

that enable the host to begin operating. In this regard, the parameters may include a URL of an initial web page (see col. 5, lines 20-34).

However, Applicants note that nowhere in this text (or the remainder of Kasso) is there any description, implicit or explicit of a declaration of a component module control object. Additionally, there is no description, implicit or explicit, that the boot image contains a script block that has a method that accesses the control object. Instead, Kasso merely teaches a boot image that contains a program that is loaded and a URL of a web page. The Patent Office may assert that the program is equivalent to (or inherently includes) the declaration of the component module control object. However, no such inherency is suggested or apparent from Kasso. In addition, if the program is equivalent or includes the control object, where is the script block that access the object?

Instead of the boot image including a script block, Kasso merely describes a boot image that includes parameters that enable the host to begin operating. Thus, Kasso merely provides that parameters enable the host to begin operating. The only example parameter given is that of a URL for an initial Web page. Accordingly, the parameters are not scripts or method calls whatsoever. Instead, the parameters are merely a set of values that can be used by the host. Such parameters/values are not and cannot be equivalent to the specifically claimed script block that comprises a method that accesses the control object. In this regard, Kasso completely fails to teach the boot strap file as claimed.

In addition to the differences between the contents of the claimed bootstrap file and that of Kasso, Kasso's boot image is merely a routine or sequence of events that are executed when a computer is powered on. Such a power-up sequence is completely distinguishable and not even remotely similar to a bootstrap file that is received in response to requesting a Web page. The present claims do not recite nor do they even remotely resemble a computer power-up sequence.

While the cited references fail to teach the claimed bootstrap file, the references also fail to teach that the control object also contains/comprises one or more resources that are used to extract a Web page. To teach the resources and the use of the resources to extract the Web page, the Office Action relies on Ko (col. 4, lines 30-33 and col. 4, lines 57-63). Col. 4, lines 30-33 merely provides that a request to download an object x2installer1 is received, and then transmitted. Such text does not even remotely describe that the x2installer1 object has resources, or more specifically, resources that are used to extract a Web page. Col. 4, lines 57-63 provides:

If not native, in step 303, the user's personal computer 100 is commanded by directive CODEBASE in the program default.htm to download a file named "x2inst.cab" from the URL "http://host.sub.13.server_address/cab/". Once downloaded, in step 304, file "x2inst.cab" is decompressed and installed as "x2inst.dll" on the user's personal computer 100. File x2inst.dll is installed according to the parameters of file x2inst.inf (Appendix B), which is provided with "x2inst.cab".

This text merely describes that if a declared object is not native (i.e., not recognized and installed) to the personal computer, a new cab file is downloaded and installed (see col. 4, lines 50-65). As can be seen, there is no indication that the downloaded file contains a resource that the Web page is based on. Instead, a cab file is downloaded from a URL address, decompressed, and installed on a computer. Such a cab file is not equivalent to a control object as claimed (i.e., an object oriented object that is declared in a bootstrap file). As defined at <http://www.techweb.com/encyclopedia/defineterm?term=CABFILE&exact=1>, a cab file is a cabinet file - a file format from Microsoft used to hold compressed files on its distribution disks. Further, the contents of the cab file must be extracted.

In view of the above definition, Applicants note that there is no indication (in either the definition or the cited text) that the cab file contains any resources that are used when extracting a Web page. Further, there is no indication in the cited text (or the remainder of Ko) that script block is used to extract a Web page based on the resources in the control object. Instead, the cab file would need to be extracted. However, as indicated in col. 4, when decompressed, the cab file contains and is installed as a dll (a dynamic link library). There is no indication that the dll is equivalent to a Web page as claimed. In this regard, a dynamic link library is not a Web page. Instead, a dll is an executable program module in Windows that performs one or more functions at runtime (see <http://www.techweb.com/encyclopedia/defineterm?term=dll>). Such a program is not a Web page in any way, shape or form. In addition, there is no suggestion that the Ko's dll contains a web page or a resource for a web page. Further, regardless of whether the ActiveX object is native or not, there is no description of the claimed resources, implicitly or explicitly.

Accordingly, Applicants submit that neither Ko nor Kasso teach, disclose, or suggest various claim elements, either implicitly or explicitly. Further, even when combined, the references teach away from Applicants' invention. For example, the combined references would teach using a bootstrap file when a computer is powered-up to properly configure a host computer to execute Java programs. Thereafter, once web page code is received, the code is analyzed to determine if an

object in the code is native. If not native, various programs such as a cab file (in the form of a dll) are downloaded and installed. Such a teaching does not even remotely describe a bootstrap file, a script block, resources in an object, or resources that are used by a method of a script block to extract a Web page.

In addition, Applicants note that instead of displaying a Web page as claimed, the method and purpose of Ko is directed towards tailoring the appearance of a web browser and is not directed towards a Web page itself (see title and Abstract of Ko).

Moreover, the various elements of Applicants' claimed invention together provide operational advantages over Ko, Kasso, and Estrada. In addition, Applicants' invention solves problems not recognized by Ko, Kasso, and Estrada.

Thus, Applicants submit that independent claims 1, 9, 17, and 25-27 are allowable over Ko, Kasso, and Estrada.

In addition to the above, the dependent claims provide additional limitations that are not even remotely suggested by the cited references. For example, dependent claims 4, 12, and 20 specifically provide that when a connection to a server is not available, the Web page is obtained exclusively via the bootstrap file that is stored locally. In teaching these claims, the Office Action relies on Estrada. Estrada merely provides for offline features that allows a user to work disconnected. However, there is no description of a bootstrap file (which is specifically relied upon in these claims). In this regard, the Web page is obtained exclusively via the bootstrap file that is stored locally. The absence of any description of such a bootstrap file in Estrada precludes the ability to read Estrada onto the present claims.

Further, dependent claim 5 provides that if a connection to a server is available, the Web page comprises content from resources of the control object and the server. Again, both Ko and Kasso fail to teach the resources as claimed.

Lastly, claim 6 provides that the resources are accessed using the RES protocol. The term RES is in capital letters in the claims and is specifically referred to on page 14, lines 10-page 15, line 9 of the present specification:

To address external images or media, the Web browser 108 may utilize a resource or "RES" protocol. The RES protocol provides a mechanism for accessing HTML pages embedded as a resource in a DLL (dynamic link library) file or Win32 executable module. Accordingly, using the RES protocol, an image included in the payload may be accessed and extracted for display by the Web browser 108.

The following example illustrates a method (using the RES protocol) for writing images and/or other media, enables the compliance with filename restrictions, and maintains expected behavior of a Web browser 108:

```
<script>
    // some common functions
    function dw(x) {document.write(x) }
    function IR(x) {return "res://InstFred.ocx/IMG/" + x}
</script>

then
    <script>dw("")</script>
The above example illustrates the use of a script to execute various functions and allows the stubbing
of IR to return only x (to aid in debugging).
```

Accordingly, the RES protocol is a specific type of protocol that is described and set forth in the specification. In rejecting claim 6, the Office Action merely cites Ko col. 7, lines 35-40. This portion of Ko merely describes FTP and HTTP. There is no mention, implicit or explicit to the RES protocol. In this regard, neither FTP nor HTTP access HTML pages embedded as a resource in a DLL file or Win32 module. Accordingly, the cited references completely fail to even remotely teach this claim.

Further, dependent claims 2-8, 10-16, and 18-24 are submitted to be allowable over Ko, Kasso, and Estrada in the same manner, because they are dependent on independent claims 1, 9, 17, and 25-27, respectively, and thus contain all the limitations of the independent claims. In addition, dependent claims 2-8, 10-16, and 18-24 recite additional novel elements not shown by Ko, Kasso, and Estrada.

III. Conclusion

In view of the above, it is submitted that this application is now in good order for allowance and such allowance is respectfully solicited. Should the Examiner believe minor matters still remain that can be resolved in a telephone interview, the Examiner is urged to call Applicants' undersigned attorney.

Respectfully submitted,

GATES & COOPER LLP
Attorneys for Applicant(s)

Howard Hughes Center
6701 Center Drive West, Suite 1050
Los Angeles, California 90045
(310) 641-8797

Date: May 25, 2004

By: 

Name: Jason S. Feldmar

Reg. No.: 39,187

JSF/amb